Score following in practice

Miller Puckette and Cort Lippe

IRCAM, 31, rue St-Merri, 75004 Paris, FRANCE
msp@ircam.fr, lippe@ircam.fr

ABSTRACT

Score following was first presented at the 1984 ICMC, independently, by Barry Vercoe and Roger Dannenburg. By 1987, IRCAM was incorporating score following in concert productions. To date, ten IRCAM pieces have relied on score following to synchronize live electronics with flute, clarinet, percussion, and piano. We noticed quickly that algorithms which work well in theory, or even in the laboratory, often cease working when confronted with real examples of instrumental writing. In the case of the clarinet and the flute it has been necessary to use acoustic-only pitch detection, which has further complicated the score following problem. The solutions we have found to date seem far from definitive, but it now seems a good moment to describe carefully and in detail what we are doing, before launching another foray into the unknown. This paper describes, in detail, the algorithm we now use, as well as the compromises that composers are still having to make in order to avoid certain of the algorithm's pitfalls.

The work reported here is part of a long-term research program at IRCAM that aims to extend the possibilities of live electronic music performance. As part of this program, several generations of digital synthesis hardware have been developed. The machine in current use for music production is the IRCAM Signal Processing Workstation (ISPW) (Lindemann 1991), which recently replaced the 4X machine as the predominant real-time signal processor at IRCAM.

Score following (Vercoe 1984; Dannenburg 1984, 1988) is the process of tracking live players as they play through a pre-determined score, usually for the purpose of providing an automatic computer accompaniment to a live player. The procedure for score following is rather simple: enter a score into a computer in some form, and then play the score as real-time input to the computer (either via a MIDI interface or a microphone and acoustic analysis), comparing the computer's stored score to the musician's playing on a note by note basis. If the comparison being made between the two is successful, the computer advances in the data base (the stored score) in parallel with the player, triggering electronic events at precise points in the performance score. In theory, one or many parameters can be followed, but with one interesting exception, we have settled on pitch. The most difficult aspect of automatic accompaniment, the necessity to extrapolate tempo in order to choose the timing of events, does not present a problem here -- it will be seen below that we can use a far simpler score following algorithm than is needed for tempo-tracking.

The score following algorithm's state consists of a pointer to the "current" note, and a set of pointers to prior notes which have not been matched (the "skip list"). There are two free parameters: the "skip number" (a non-negative integer) and the "skip time" (in units of time.) The "current note" is the note immediately after the furthest note (in the ordering of the data base) that has been matched. The score follower is started by setting the current note to the first note of the data base. Once the last note has been matched, the current note is undefined (but matching can still continue if there are notes in the skip list.) When a live note is played, the follower attempts to match it with a note in the data base; the matched note must have the same pitch as the live one (alternatively, if the live notes might have octave errors such as from a pitch tracker, matches can be accepted between notes that differ by an octave.) First, a match is sought from the skip list. For the match to be valid, the skipped note may not be more than the skip time prior to the current note (the time being taken from the data base); that is, the

follower will not back up more than that amount. (We can thus drop old notes from the skip list to control its size.)

If no match is found from the skip list, the search is continued starting at the current note. First, a fixed number of notes are examined, given by the skip number. Then, starting from the last such note, the search is continued for all notes that lie within the skip time. If a match is found at or past the current note, the current note and skip list are updated. In figure 1, a possible score follower state is shown, indicating which notes are matchable at a given time. Typical choices are one or two for the skip number, and .1 to .3 seconds for the skip time. The skip number is primarily relevant in slow, melodic passages where no two notes might be within the skip time of each other in the data base; it represents the number of mistakes of omission the player may make before the follower gets lost. (From the point of view of the score follower, the player "jumps ahead" to the next correct note that is played.) The skip number is best kept small because a wrong or extra note can cause the follower to jump forward; the greater the skip number, the greater the number of possibilities for matching the incorrect note.

In contrast, the skip time is most significant for vertical structures such as a chord. A chord is not likely to be played in the same order every time; the skip time controls how far the score follower is willing to go back and forth in the data base to "collect" all the notes of a chord as they come in. As with the skip number, the skip time is best kept small. The score follower pays attention to the beginnings of notes only; their durations, both in the score data base and in the live performance, are ignored.

Score following may be done either on a Macintosh or the ISPW, using the MAX program (Puckette 1988). On the Macintosh, MAX uses the MIDI standard to control either commercial synthesizers or the 4X; on the ISPW, synthesis and sound processing are also done within MAX. An editor, EXPLODE, has been written for MAX to facilitate the maintenance of the score data base. A more complete description of EXPLODE appeared in (Puckette 1990). For score following, incoming events are sent to EXPLODE, and conditionally generate events on EXPLODE's output. EXPLODE regards channel 1 as the set of events to follow, and all other channels as global actions. Incoming events which match an event on channel 1 cause the matched event to be output. The matched events need not come out in exactly the same order as they occur in the sequence; moreover, if an event is skipped in the input it never appears on the output. Events whose channels are not 1 are output as soon as they are passed; i.e., as soon as a later event from channel 1 is matched.

A musical example of score following for triggering signal processing events is the fifty-minute piece, "Pluton", by Philippe Manoury. This piece, originally for piano and 4X, has since been ported to the ISPW. The function of the signal processing in "Pluton" is primarily to transform the sound of the piano, via harmonizing, frequency shifting, time-stretching, spatialization, reverberation, and analysis/resynthesis. An example is shown in figure 2, which uses an infinite reverb and two samplers. The circled numbers 2, 3, and 4 represent global actions: action 2 sets a reverberator to infinite reverb time and sends the piano sound to it for about .3 seconds (after waiting .2 seconds to avoid having the attack of the piano note in the reverb.) Actions 3 and 4 record the two chords of the piano into the two samplers (represented by the top two staves in the figure), and start note-generation processes which replay the chords with an irregular rhythm (further actions replace the samples and change the rhythm of playback.) The piano part of the example is somewhat free rhythmically, thus there would be little value in trying to extract a tempo from it.

If the performance and the computer score differ, this may produce undesired results. Even if the score follower always works in rehearsals, a musician is not infallible. It is essential that someone be on hand to follow both the musician's playing and the computer's following during a performance, ready to intervene if and when the performer and computer fall out of synchronization. Also, composers are often forced to make compromises so that their music is followed in such a way that the electronic events in the score are correctly triggered. These compromises fall into three categories: 1.) Writing which cannot be accurately followed because of the nature of the written material: trills, tremolos, repeated notes, multiphonics, and secondary sound sources. 2.) Passages which are of great interpretive difficulty or passages which may be error-prone due to

performer nervousness. 3.) Signal processing versus score following ordering problems.

Figure 3, from the composition "Music for Clarinet and ISPW" by the second author, exemplifies one major difficulty in using the algorithm described above. Composers often ask players to play passages at the limits of their ability, and the players often make strings of errors in fast passages. Also, any acoustic pitch follower will make more errors in fast passages than in slow ones. A tempo-sensitive score follower of the sort described in (Vercoe 1985) tends to handle such situations transparently, but in our case it is often necessary to reduce the computer score to a subset of the performer score, containing only certain, more reliable notes.

In Figure 4 (taken from "Explosant-Fixe" by Pierre Boulez), we see a succession of tremolos, with common notes between them. Since the tremolos are over wide intervals, the flute tends to produce extra notes in the transition, some being audible, and others resulting from transients which are not stable enough to be heard. It is unknown in advance, of course, how long or how fast the tremolos will be played (except, of course, by guessing from a previously measured tempo). The compromise taken here was to add certain grace notes between the tremolos, which the follower is forced to await before proceeding. The tremolos themselves are entered as a single pair of notes, since duplicating the notes of the tremolo will not in any way improve the matching.

It is sometimes desired that the score follower await a specific input without the possibility of jumping forward. This is accomplished by adding artificial "forcing" notes, out of the range of the instrument, so that the algorithm cannot jump over the group without visiting the desired input.

Figure 5, by the second author again, presents the worst quandary of all. In effect, the clarinet plays a continuous 'b' below middle 'c', which becomes breathy in the middle and returns to normal at the end. Computer responses are desired at the beginning, middle, and end of the phrase. In this register the grace notes are not always picked up by the pitch tracker, and the "concert effect" ensures that they are least likely to be properly reported when they are most desired. The timing is free, so tempo awareness would not help. This is something a human accompanist has no trouble following; but any solution so far proposed for computer involves some kind of cheating.

It will have occurred to the reader that all of the shortcomings of this score follower could be overcome by the addition of some rudimentary "intelligence", particularly taking advantage of timing information from the input. So far, we have hesitated to do so. The algorithm described may be comprehended and used by a non-computer-scientist. An "intelligent" algorithm might fail less often, but in worse or less easily repeatable ways. If a particular spot in a score works in rehearsal, it is likely to work in performance too. Finally, if a passage seems questionable in rehearsal, it is usually possible to play it out of tempo in order to reproduce and correct the problem.

REFERENCES

R. Dannenburg, "An On-line Algorithm for Real-Time Accompaniment," Proceedings, ICMC, (Paris, France), pp. 193-198, 1984.

R. Dannenburg and H. Mukaino, "New Techniques for Enhanced Quality of Computer Accompaniment," Proceedings, ICMC, (Cologne, Germany), pp. 243-249, 1988.

E. Lindemann et al., "The Architecture of the IRCAM Music Workstation," Computer Music Journal vol. 15(3), pp. 41-49, 1991.

M. Puckette, "The Patcher," Proceedings, ICMC, (Cologne, Germany), pp. 420-429, 1988.

M. Puckette, "EXPLODE: A User Interface for Sequencing and Score Following," Proceedings, ICMC, (Glasgow, Scotland), pp. 259-261, 1990.

B. Vercoe, "The Synthetic Performer in the Context of Live Performance," Proceedings, ICMC, (Paris, France), pp. 199-200, 1984.

B. Vercoe and M. Puckette, "Synthetic Rehearsal: Training the Synthetic Performer," Proceedings, ICMC, (Vancouver, Canada), pp. 275-278, 1985.

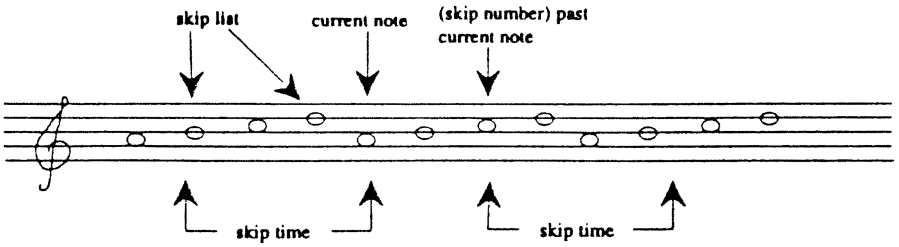


Figure 1. A "score" (the phrase a-b-c-d repeated three times), indicating a possible state for the score follower. Incoming notes can match either an element of the skip list, or a note on or past the "current note", but not past the end of the second "skip time" interval (the tenth note in this example). At this point, the notes b and d would be eaten by the skip list; the notes a and c would match the fifth and seventh notes, respectively.

Figure 2. (Manoury)

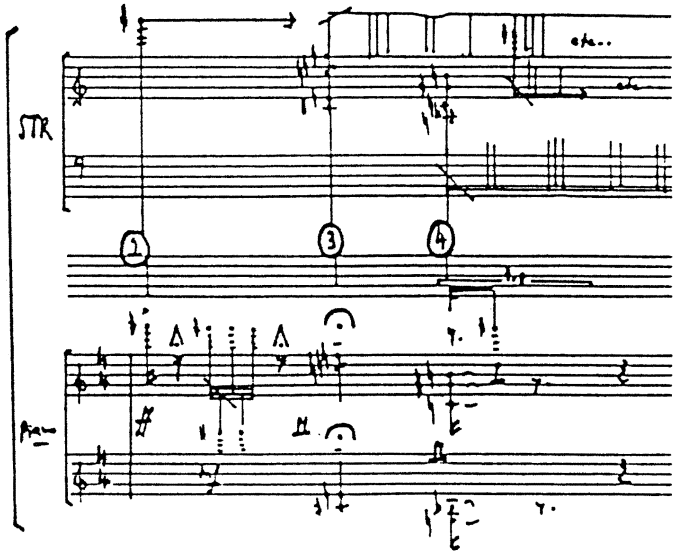
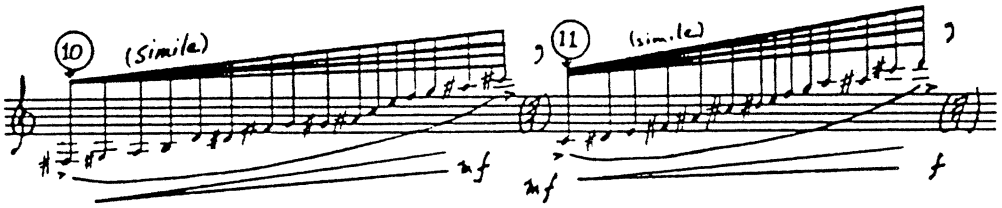


Figure 3. (Lippe)



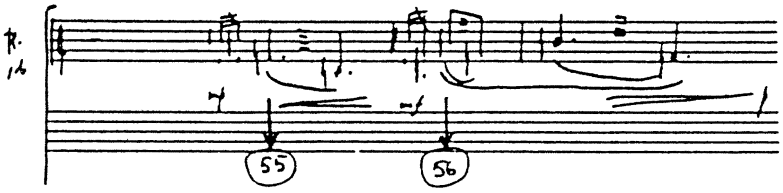


Figure 4. (Boulez)

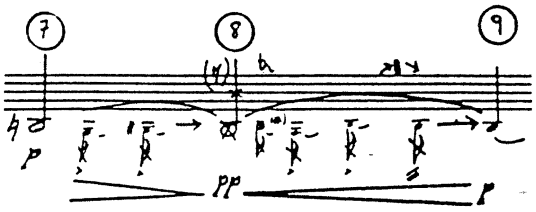


Figure 5. (Lippe)